

What is Claimed:

1. A method for initiating a bifurcated process to be debugged by a debugger, where said process is scheduled using a shadow process running in a first execution environment and where said bifurcated process is executed using a trusted agent process running in a second execution environment, said method comprising:
 - initiating said shadow process in debug mode in said first execution environment;
 - initializing said trusted agent process in said second execution environment;
 - attaching said debugger to said shadow process;
 - once said debugger is attached, allowing said shadow process to schedule execution of said process by said trusted agent.
2. The method of claim 1, where said step of initiating said shadow process comprises sending an agent image to said second execution environment.
3. The method of claim 1, where said step of initiating said shadow process comprises sending notification that said process should be debugged to said second execution environment.
4. The method of claim 1, where said step of initiating said shadow process comprises sending a unique identifier of said shadow process to said second execution environment.
5. The method of claim 1, where said step of initiating said shadow process in debug mode in said first execution environment comprises:
 - deferring the scheduling of execution of said process by said trusted agent.
6. The method of claim 1, where said step of initializing said trusted agent process in said second execution environment comprises laying out process memory structures in memory.
7. The method of claim 1, where said step of initializing said trusted agent process comprises sending a unique shadow process identifier for said shadow process to said second execution environment.
8. The method of claim 7, where said unique shadow process identifier comprises a process environment block (PEB).

9. The method of claim 7, where said unique shadow process identifier is used to verify an identity of said process when a request regarding process memory is sent by said debugger to said second execution environment.
10. The method of claim 7, where said step of initializing said trusted agent process further comprises sending a unique shadow thread identifier corresponding to each shadow thread of said shadow process to said second execution environment.
11. The method of claim 10, where said unique shadow thread identifier comprises a kernel thread (KTHREAD) pointer.
12. The method of claim 10, where said unique shadow thread identifier is used to verify an identity of trusted agent thread corresponding to said shadow thread corresponding to said unique shadow thread identifier when a request regarding thread context of said trusted agent thread is sent by said debugger to said second execution environment.
13. The method of claim 1, where said method further comprises:
determining whether debugging should be permitted on said bifurcated process, and
aborting debugging of said bifurcated process if said determination concludes that debugging should not be permitted.
14. The method of claim 13, where said step of determining whether debugging should be permitted comprises:
determining whether a debugging flag is set for said bifurcated process.
15. At least one of an operating system, a computer readable medium having stored thereon a plurality of computer-executable instructions, a co-processing device, a computing device, and a modulated data signal carrying computer executable instructions for performing the method of claim 1.
16. A method for debugging a bifurcated process, where said process is scheduled using a shadow process comprising at least one shadow thread running in a first execution environment and where said bifurcated process is executed using a trusted agent process running in a second execution environment, where said trusted agent process comprises trusted agent threads, each of said trusted agent threads corresponding to one of said shadow process threads, said method

comprising:

- receiving a request message from said debugger regarding a resource;
- verifying an identity of said requested resource from said request message;
- performing the request in said request message; and
- responding to said request message with the result of said request.

17. The method of claim 16, where said steps of receiving a request message from said debugger regarding a resource, verifying the identity of said requested resource from said request message, performing the request in said request message, and responding to said request message with the result of said request are performed by an admin thread running in said second execution environment.

18. The method of claim 17, where said admin thread is owned by said bifurcated process.

19. The method of claim 16, where said request message comprises a request selected from among the following group: a request to get thread context for one of said trusted agent threads; a request to set thread context for one of said trusted agent threads; a request to read process memory for said trusted agent process; and a request to write process memory for said trusted agent process.

20. At least one of an operating system, a computer readable medium having stored thereon a plurality of computer-executable instructions, a co-processing device, a computing device, and a modulated data signal carrying computer executable instructions for performing the method of claim 16.

21. A system comprising:

- a first execution environment that comprises a scheduler, said first execution environment being adapted to schedule and dispatch a plurality of threads for execution on a processor, said first execution environment being further adapted to initiate a shadow process in debug mode in said first execution environment, to attach a debugger to said shadow process, and to, upon receipt of an indication that execution should begin, to schedule at least one thread of said shadow process.

- a second execution environment that is adapted to dispatch a plurality of second threads for execution on said processor, said second execution environment being further adapted to initialize a trusted agent process corresponding to said shadow process in said second

execution environment, and to, upon completion of said initialization, send an indication that execution should begin to said first execution environment.

22. The system of claim 21, where said initiation of said shadow process comprises sending an agent image to said second execution environment.

23. The system of claim 21, where said initiation of said shadow process comprises sending notification that said process should be debugged to said second execution environment.

24. The system of claim 21, where said initiation of said shadow process comprises sending a unique identifier of said shadow process to said second execution environment.

25. The system of claim 21, where said initiation of said shadow process in debug mode in said first execution environment comprises deferring the scheduling of execution of said process by said trusted agent.

26. The system of claim 21, where said initiation of said trusted agent process comprises sending a unique shadow process identifier for said shadow process to said second execution environment.

27. The system of claim 26, where said unique shadow process identifier comprises a process environment block (PEB).

28. The system of claim 26, where said unique shadow process identifier is used to verify an identity of said process when a request regarding process memory is sent by said debugger to said second execution environment.

29. The system of claim 21, where said initiation of said trusted agent process further comprises sending a unique shadow thread identifier corresponding to each shadow thread of said shadow process to said second execution environment.

30. The system of claim 29, where said unique shadow thread identifier comprises a kernel thread (KTHREAD) pointer.

31. The system of claim 29, where said unique shadow thread identifier is used to verify an identity of trusted agent thread corresponding to said shadow thread corresponding to said

unique shadow thread identifier when a request regarding thread context of said trusted agent thread is sent by said debugger to said second execution environment.

32. The system of claim 21, wherein said first execution environment comprises a first operating system, and wherein said second execution environment comprises a second operating system.

33. The system of claim 21, wherein the first execution environment is expected to conform its behavior to a first specification, wherein said second execution environment is expected to conform its behavior to a second specification, and wherein the expectation that the second execution environment will behave according to the second specification is relatively greater than the expectation that the first execution environment will conform its behavior to the first specification.

34. The system of claim 21, where said second execution environment further determines whether debugging should be permitted on said bifurcated process, and aborts debugging of said bifurcated process if said determination concludes that debugging should not be permitted.

35. The system of claim 34, where said determination is based on whether a debugging flag is set for said bifurcated process.

36. A system comprising:

a first execution environment that comprises a scheduler, said first execution environment being adapted to schedule and dispatch a plurality of first threads for execution on a processor, said first execution environment running at least one shadow process comprising at least one shadow process thread, where a debugger is attached to said shadow process;

a second execution environment that is adapted to dispatch a plurality of second threads for execution on said processor, said second execution environment running at least one trusted agent, each of said at least one trusted agent corresponding to one of said at least one said shadow process, where each of said at least one trusted agent comprising at least one trusted agent thread, where each of said trusted agent threads corresponds to one of said shadow process threads, where a bifurcated process comprises said shadow process and said trusted agent and is scheduled using said shadow process and executed using said trusted agent, and where said second execution environment is adapted to (a) receive a request message from said debugger regarding a resource; (b) verify an identity of said requested resource from said request message;

(c) perform the request in said request message; and (d) respond to said request message with the result of said request.

37. The system of claim 36, where said request message comprises a request selected from among the following group: a request to get thread context for one of said trusted agent threads; a request to set thread context for one of said trusted agent threads; a request to read process memory for said trusted agent process; and a request to write process memory for said trusted agent process.

38. The system of claim 36, where said second execution environment stores unique identifier information for each possible resource for which a valid request may be made, and where said verification of an identity of said requested resource comprises determining whether said resource is one of said possible resources for which a valid request may be made.

39. The system of claim 36, where said second execution environment stores identifier information comprising unique process identifier information for said shadow process and unique thread identifier information for said shadow threads, and said verification of said requested resource from said request message comprises using said identifier information to identify the resource in said second execution environment.

40. The system of claim 36, wherein said first execution environment comprises a first operating system, and wherein said second execution environment comprises a second operating system.

41. The system of claim 36, wherein the first execution environment is expected to conform its behavior to a first specification, wherein said second execution environment is expected to conform its behavior to a second specification, and wherein the expectation that the second execution environment will behave according to the second specification is relatively greater than the expectation that the first execution environment will conform its behavior to the first specification.

42. The system of claim 41, wherein the criterion determines whether running the second thread would cause the second execution environment to behave in a manner that would violate the second specification.